The University of Texas at El Paso
Department of Computer Science
CS 3331 – Advanced Object-Oriented Programming
Code Review Checklist

Victor Herrera
Christian A. Gomez Implementation

- Does this code do what it is supposed to?

  The code is designed to serve as user ticket system for users who are members of University of Texas at El Paso and to who users that are not. Other functionality of user will be employee users which will serve as administrators that will have the option to go into the system and view records. They will also have the ability to create new events that will allow customers to purchase. The overall functionality of code meets the requirements prescribed in PA5.

- Can it be simplified?

  There have been multiple changes to the code and has been simplified from previous version. The code can still be simplified to help with time and space complexity. That is something that is still being worked on and should reach a better simplification level.

- Is the code dynamic or hardcoded?

  The code is dynamically code and meets the requirements. The code has been tested and handles exceptions and invalid inputs. If a new csv file is given, with the same header and with the name in different indexes the code is set up to handle this.

- Is the code maintainable?

  The code is maintainable there is plenty of documentation within the JavaDocs to provide clarification if any questions should arise of what the methods do.

Logic

- Cases where code does not behave as expected/intended?

  In automatic purchasing and Administrator options. In automatic purchasing threw an exception array index out bounds. Something that was not expected. In Administrator options the code would continue with invalid input. When an administrator would view event with invalid input the program would continue to function. The end action would not take place, example the event requested would not print to screen. There was no error, printed or exception thrown but this needs to be modified so program does not continue to function with invalid input.

- Test cases where it may fail?

  As of now all modifications have been made and only a couple cases fail. One specific is the date time input from user. If administrator decides to enter a time for am. The program handles that as an invalid input. It will request the user to enter a time again, but it must always be in pm. So, more changes need to be made.

Readability/Style

- Easy to read/understand?

  The code readability is facile and has documentation to provide any additional details should questions arise.

- What parts can be modified or adjusted?
  As mentioned in testing test cases. The method that handles date and time needs to be modified to handle all cases not just pm input.

- Is the structure appropriate?

  The structure of the code is appropriate and set up with appropriate classes for functionality.
- Does it follow the appropriate language style?
  The code follows the Java Syntax.
- Is the code well documented?

  There is enough documentation within the Javadoc's to provide an explanation of any uncertainty within the code.

Performance
- What is the code complexity?
  The code is not difficult to understand some areas need to be modified, in order to be less complex. The use of the buffer instead of memory is something that is being looked at to improve space and time. Currently the time complexity of the code is not at its best performance but can be improved. Current time complexity is squared.
- How does the complexity change with various inputs? The complexity of the code does change, with different inputs. The change is not so drastic that is more costly than efficient when it comes to time and space. It continues to be reviewed to improve efficiency.